Regular Paper

Int. J. on Recent Trends in Engineering and Technology, Vol. 9, No. 1, July 2013

Performance Analysis of Steepest Descent Decoding Algorithm for LDPC Codes

Finney Daniel Shadrach S [1], Michaelraj Kingston R [2], and Dr. Ramesh J [2]

[1] PSG College of Technology/Department of ECE, Coimbatore, India

Email: finneydaniels@gmail.com

[2] PSG College of Technology/Department of ECE, Coimbatore, India

Email: king.pane@gmail.com, jramesh60@yahoo.com

***Abstract* - Among various hard decision based Bit Flipping (BF) algorithms for decoding Low-Density Parity-Check (LDPC) codes such as Weighted Bit Flipping (WBF), Improved Reliability Ratio Weighted Bit Flipping (IRRWBF) etc., the Steepest Descent Bit Flipping Algorithm (SDBF) achieves better error performance. In this paper, the performance of the Steepest Descent Algorithm for both single steepest descent and Multi steepest descent modes is analysed. Also the performance of IEEE 802.16e standard is analysed using Steepest Descent Bit Flipping (SDBF) decoding algorithm. SDBF requires fewer check node and variable node operations compared to Sum Product Algorithm (SPA) and Min Sum Algorithm (MSA). The SDBF achieves a coding gain of 0.1 ~ 0.2 dB compared to Single-SDBF without requiring complex log and exponential operations.**

***Index Terms* —Bit Flipping Algorithms, Hard Decision Decoding, LDPC Codes, Steepest Descent, IEEE 802.16e.**

I. Introduction

Low density parity check codes were introduced by Gallager [1] in 1962. Mackay and Neal reintroduced LDPC code in 1996[2]. Due to their excellent error performance, LDPC codes have achieved significant attention and have been adopted by many recent communication standards such as 10 Gigabit Ethernet (10GBASE-T), digital video broadcasting (DVB-S2) and WiMax (802.16e). The IEEE WiMax standard [6] covers a large range of wireless transmission applications. Compared to Wi-Fi (or Wireless LAN), it can support high throughput over larger distances, even with higher mobility involved. The upcoming IEEE WiMax 802.16e standard, also referred to as Wireless-MAN [7], is the next step toward very high throughput wireless backbone architectures, supporting up to 70Mbps. The WiMax standard features LDPC codes as an optional channel coding scheme.

LDPC codes belongs to a family of linear block code in which non-zero entries (1's) will be sparse. LDPC codes can be decoded by an iterative decoding algorithm. It has been shown that these codes achieve a remarkable performance with iterative decoding that is very close to the Shannon limit [1], Mackay et.al [2]. Consequently, these codes have become strong competitors to turbo codes for error control in many communication and digital storage systems where high reliability is required. The decoding is an iterative process which exchanges information between two types of nodes. The WiMax LDPC code was designed with respect to hardware constraints, however, the high number of defined codes (code rates and codeword Sizes) specified by different members of the WiMax consortium imposes significant challenges on an LDPC decoder realization.

LDPC codes can be decoded using various decoding schemes such as hard -decisions, soft -decisions and hybrid decoding schemes.

Bit flipping algorithms is a hard decision scheme which possesses a good trade -off between error-correcting performance and decoding complexity compared to Belief Propagation Algorithm (BPA) and its variants such as Sum-Product and Min-Sum algorithms (MSA). The high-complexity Sum-Product algorithm (SPA) was shown to achieve a near - capacity performance. However, the Steepest Descent bit-flipping algorithm strikes a good trade -off between the associated decoding complexity and the achievable performance. The attractive property of this algorithm is it significantly lower decoding in comparison to the SPA.

LDPC codes can be described in two forms [10]: the matrix form and the graphical form. The graphical form was introduced by Tanner and called as tanner graph. Fig.2 represents the tanner graph for corresponding parity check matrix shown in Fig.1. The Tanner graph is a bipartite graph with two types of nodes called variable nodes and check nodes. The connection between check nodes and variable node is done using the H matrix, in such a way that columns of H is considered as the variable nodes and rows of H is considered as check nodes. In any location there is a '1' in the matrix, the related variable and check nodes will be connected by an edge. The important parameter that is considered in tanner graph is its girth. Girth is the length of the shortest loop that begins and ends in the same node. Usually longer girth in tanner graph provides efficient code. So, avoiding the girth of length 4 and 6 is essential. LDPC code can be of two types regular and irregular matrix. A LDPC code is called regular if wc is constant for every column and wr = wc · (n/m) is also constant for every row. The example matrix shown in Fig1 is regular with wc = 2 and wr = 4. It's also possible to see the regularity of this code while looking at the graphical representation. There is the same number of incoming edges for every variable node and also for all the check nodes. If H is low density but the numbers of 1's in each row or column aren't constant the code is called an irregular LDPC code.

DOI: 01.IJRTET.9.1.18

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Fig.1: Example of a regular (8, 4) matrix

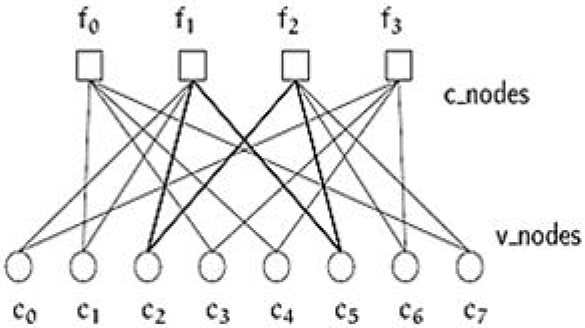

Fig.2: Tanner graph representation for (8, 4, 2) matrix shown in Fig 1

The rest of this paper is organized as follows. In section II , a brief overview of decoding algorithm is explored. Section III presents the code description of IEEE802.16e. Section IV provides simulation results before concluding remarks in Section V.

II. Literature Survey

Assume a binary (M,N) parity check matrix H, where M is the number of parity check equations ie., rows and N is the number of bits ie., columns. The non-zero elements in row i of H N *(i)* and the non-zero elements in column 5ØWÜ is (5ØWÜ). The i-th bipolar syndrome of x is then written as $\prod_{j \in N(i)} x_j$. Let C be the codeword of a regular (d_v, d_c) LDPC code of length N, defined as the null space of a sparse parity check matrix $H = [h_{m,n}]$. We assume information transmission using BPSK modulation over an AWGN channel, which maps a codeword c = ($c1$, $c2$, ..., cN) into a bipolar sequence x =($x1$, $x2$, ..., xN), according to $x_n = 2C - 1$, $n \in [1, N]$. Let y = *(Y1, Y2, ..., YN)* be the soft-decision received vector and x be represent the vector of received samples and x $\in${+1,-1} be the hard detected bits. Let the received real value sequence is $y_n = x_n + w_n$, where w_n statistically independent Gaussian random variables.

In Fig 3, a pictorial representation of the receiver shows two different blocks: block A represents the set of all variable nodes, or variable node detector (VND); block B represents the set of all check nodes, or check node detector (CND). Each message set, at the input or output of a block, can be viewed as the output of a channel with the actually transmitted codeword at its input: the message set is indeed a stochastic function of the transmitted codeword. For each message set, the Mutual Information (MI) between the transmitted codeword and the message set can then be computed. In Fig 3, Iv denotes the MI at the output of the VND, and Ic the MI at the output of the CND. The underlying

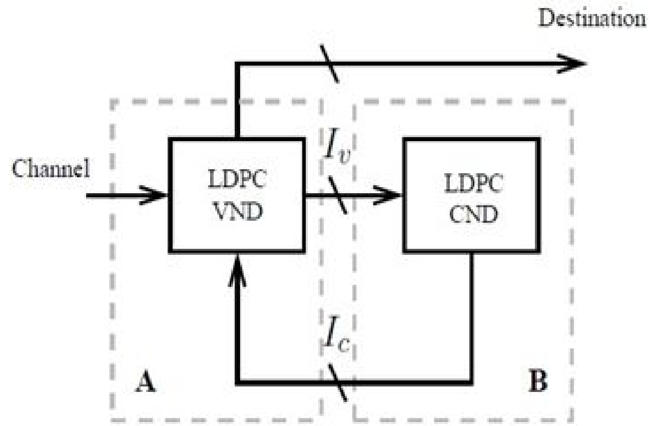

Fig.3: General Block Diagram of Decoder

assumption is that the MI at the output of a block is a function of the MI at the input, regardless of the actual statistical distribution of the messages. The decoding process can thus be described as a recursive computation of the MI between variable node blocks (LDPC VND) and check node blocks (LDPC CND).

A. Bit Flipping Algorithm

In a decoding process of a BF algorithm, bit flip positions are determined based on the values of an inversion function.

The inversion functions of Weighted Bit Flipping (WBF) is defined by Zhang and Fossorier [3], Jiang et.al [4], and it is given in (1) as follows;

$$E_n^{(WBF)} = \sum_{m \epsilon M(n)} (2S_m - 1)\, |y|_{min-m} \tag{1}$$

The inversion function of WBF provides a measure of the invalidness of symbol assignment on 5ØeÜ, which is given by the sum of the weighted bipolar syndromes. It initially finds the most unreliable message node participating in each individual check. The Modified WBF (MWBF) algorithm considers not only the reliability of the syndrome sequence, but also the reliability information of the code bits. The MWBF is given as,

$$E_n^{(MWBF)} = \sum_{m \epsilon M(n)} (2S_m - 1)|y|_{min-m} - \alpha |y_n| \tag{2}$$

Although the inversion function of MWBF has a similar form to the inversion function of the WBF algorithm, the inversion function of the MWBF algorithm consists of an additional term αy_n where α is a positive real number and y_n is the information of the corresponding bit received from the channel.

Both the WBF and the MWBF algorithms consider only the specific check-node based information, which relies on the message node having the lowest soft-value . The Reliability Ratio Weighted Bit Flipping (RRWBF) is defined as Guo and Hanzo [5] ,

$$E_n^{(RR-WBF)} = \sum_{m \epsilon M(n)} \frac{2S_m - 1}{R_{mn}} \tag{3}$$

DOI: 01.IJRTET.9.1.18

Where R_{mn} is the reliability ratio. The reliability ratio algorithm normalizes the value of E_n at each bit. The Gradient Descent algorithm is composed of the correlation between the hard decision and the received soft value of the corresponding bit plus the syndrome sum as follows Wadayama et.al [6]:

$$E_n^{(GD)} = x_k y_k + \sum_{i\epsilon M(k)} \prod_{j\epsilon N(i)} x_j \tag{4}$$

B. Steepest Descent Bit Flipping Algorithm

The Maximum Likelihood (ML) decoding problem for the binary AWGN channel is equivalent to the problem of finding a (bipolar) codeword which gives the largest correlation to a given received word. This objective function has been inspired by the correlation MLD rule. Based on this correlation decoding rule, we define the following objective function f(x) as follows:

$$f(x) = \sum_{j=1}^{n} x_j y_j + \sum_{i=1}^{m} \prod_{j\epsilon N(i)} x_j \tag{5}$$

Here the first term corresponds to the correlation between a bipolar codeword and the received word, and the second term is the sum of the bipolar syndromes of x. Here the correlation term should be maximized. The inversion function can be obtained from the objective function. The inversion function of the steepest descent bit flipping algorithm is given by,

$$E_n^{(SDBF)} = x_m y_m + \sum_{m\in M(n)} (2S_m - 1)\,|y|_{min-m} \tag{6}$$

In order to reduce the computational time we use either single or multi bit flipping. With the single bit flipping the objective function is gradually ascended but much faster convergence could be expected with a large step size. In order to obtain the faster convergence the multi BF is used. In single BF only one bits are flipped in each iterations based on the inversion function, due to which computational time may be large. But in the case of multi BF parameter such as inversion threshold is used based on which all bits can be flipped at the same time. Multi BF may not sound good for smaller codeword length and in such cases single BF can be most preferred. The inversion threshold is given as ,

$$\theta^{(SDBF)} = \frac{(\sigma^2)^2}{2} * \frac{1}{(E_{neg})} \sum_{n\in\{1,2..\}} E_n^{(SDBF)} \tag{7}$$

Here σ^2 is the variance of the received signal, E_{neg} denotes the number of negative inversions. Based on the inversion threshold the multi bit flipping is performed.

Steps in Decoding Process:

1. Intialisation :
Initialise the no.of iteration .
2. Hard Decision :
Let $x_j = sign(y_j)$, where $sign(x) =\ +1$ if $x \geq 0$, else -1 .
3. Parity Check :
If $\prod_{j\epsilon N(i)} x_j = +1$ for all i ,then stop decoding and output x
4. Bit Flipping :
Initially set $f_1 = f(x)$ and $f_2 = f(x)$, where f_1 and f_2 are the previous and present value of the objective function. Here if $f_1 \leq f_2$ and $E_n^{(SDBF)} < \theta^{(SDBF)}$, then flip all the bits. Else flip a single bit .
5. Update the algorithm :
Check the no.of iterations. If no.of iterations is greater than the maximum iterations then stop and exit. Else repeat the decoding process.

III. IEEE 802.16e Code Description

The LDPC code is based on a set of one or more fundamental LDPC codes. Each of the fundamental codes is a systematic linear block code. Each LDPC code in the set of LDPC codes is defined by a matrix H of size *m*-by-*n*, where *n* is the length of the code and *m* is the number of parity check bits in the code. The number of systematic bits is *k*=*n*-*m*. The matrix H is defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,1} & \mathbf{P}_{0,2} & \cdots & \mathbf{P}_{0,n_b-2} & \mathbf{P}_{0,n_b-1} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \cdots & \mathbf{P}_{1,n_b-2} & \mathbf{P}_{1,n_b-1} \\ \mathbf{P}_{2,0} & \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \cdots & \mathbf{P}_{2,n_b-2} & \mathbf{P}_{2,n_b-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{P}_{m_b-1,0} & \mathbf{P}_{m_b-1,1} & \mathbf{P}_{m_b-1,2} & \cdots & \mathbf{P}_{m_b-1,n_b-2} & \mathbf{P}_{m_b-1,n_b-1} \end{bmatrix} = \mathbf{P}^{H_b}$$

where $P_{i,j}$ is one of a set of *z*-by-*z* permutation matrices or a *z*-by-*z* zero matrix. The matrix H is expanded from a binary base matrix H_b of size m_b-by-n_b, where $n = z \cdot n_b$ and $m = z \cdot m_b$, with *z* an integer ≥ 1. The base matrix is expanded by replacing each 1 in the base matrix with a *z*-by-*z* permutation matrix, and each 0 with a *z*-by-*z* zero matrix. The base matrix size n_b is an integer equal to 24. The permutations used are circular right shifts, and the set of permutation matrices contains the *z*×*z* identity matrix and circular right shifted versions of the identity matrix. Because each permutation matrix is specified by a single circular right shift, the binary base matrix information and permutation replacement information can be combined into a single compact model matrix H_{bm}. The model matrix H_{bm} is the same size as the binary base matrix H_b, with each binary entry (*i*,*j*) of the base matrix H_b replaced to create the model matrix H_{bm}. Each 0 in H_b is replaced by a blank or negative value (e.g., by –1) to denote a *z*×*z* all-zero matrix, and each 1 in H_b is replaced by a circular shift size $p(i,j) \geq 0$. The model matrix H_{bm} can then be directly expanded to H. H_b is partitioned into two sections, where H_{b1} corresponds to

the systematic bits and H_{b2} corresponds to the parity-check bits, such that

$$\mathbf{H}_b = \left[\left(\mathbf{H}_{b1}\right)_{m_b \times k_b} \; \vdots \; \left(\mathbf{H}_{b2}\right)_{m_b \times m_b} \right] \qquad (8)$$

In particular, the non-zero submatrices are circularly right shifted by a particular circular shift value. Each 1 in H'_{b2} is assigned a shift size of 0, and is replaced by a $z{\times}z$ identity matrix when expanding to H. The two 1s located at the top and the bottom of h_b are assigned equal shift sizes, and the third 1 in the middle of h_b is given an unpaired shift size.

The Fig.4 shows the base matrix for the Wi-Max system. A base model matrix [8] [9] is defined for the largest code length (n=2304) of each code rate. The set of shifts $\{p(i,j)\}$ in the base model matrix are used to determine the shift sizes for all other code lengths of the same code rate. Each base model matrix has n_b=24 columns, and the expansion factor z_f is equal to n/24 for code length n. Here f is the index of the code lengths for a given code rate, f=0, 1, 2, … 18. For code length n=2304 the expansion factor is designated z_0=96. For code rates 1/2, the shift sizes $\{p(f, i, j)\}$ for a code size corresponding to expansion factor z_f are derived from $\{p(i,j)\}$ by scaling $p(i,j)$ proportionally,

$$p(f,i,j) = \begin{cases} p(i,j), & p(i,j) \le 0 \\ \left\lfloor \dfrac{p(i,j) z_f}{z_0} \right\rfloor, & p(i,j) > 0 \end{cases} \qquad (9)$$

where $\lfloor x \rfloor$ denotes the flooring function which gives the nearest integer towards ∞. The LDPC code flexibly supports different block sizes for each code rate through the use of an expansion factor. Each base model matrix has n_b=24 columns, and the expansion factor (z factor) is equal to n/24 for code length n. In each case, the number of information bits is equal to the code rate times the coded length n.

Rate 1/2:

-1	94	73	-1	-1	-1	-1	-1	55	83	-1	-1	7	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	27	-1	-1	-1	22	79	9	-1	-1	-1	12	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	24	22	81	-1	33	-1	-1	-1	0	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1
61	-1	47	-1	-1	-1	-1	-1	65	25	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1
-1	-1	39	-1	-1	-1	84	-1	-1	41	72	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	46	40	-1	82	-1	-1	-1	79	0	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1
-1	-1	95	53	-1	-1	-1	-1	-1	14	18	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
-1	11	73	-1	-1	-1	2	-1	-1	47	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1
12	-1	-1	-1	83	24	-1	43	-1	-1	-1	51	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1
-1	-1	-1	-1	-1	94	-1	59	-1	-1	70	72	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1
-1	-1	7	65	-1	-1	-1	-1	39	49	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
43	-1	-1	-1	-1	66	-1	41	-1	-1	-1	26	7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Fig.4: Base parity check matrix for rate ½ mobile Wi-Max system

IV. Simulation Results

In this section the decoding performance of SDBF algorithms and IEE802.16e using SDBF obtained from computer simulations are discussed.

For the analysis of SDBF the simulation is done in MATLAB version 7.9.0.529(R2009b). The simulation is performed for binary PEGReg504x1008 [7] which has column weight 3 and row weight 6 and it's a regular matrix. The codeword is transmitted through the AWGN channel using bpsk modulation scheme. The maximum decoding iteration is set to 20. The size of the codeword is taken as 1008 with a rate 0.5.

Fig.5 and Fig.6 shows the BER versus Eb/No analysis for both Progressive Edge Growth and Gallager constructions respectively. Simulated results shows that the better BER performance of around 10^{-5} can be achieved with less complexity and in less computational time when comparing to other BF algorithms [3-6]. The Table 1 shows the SNR value corresponding to BER considering various parameters such as code rate, maximum iteration etc. A considerable coding gain of around 0.1 ~ 0.2 is achieved. The IRRWBF and Single-SDBF holds good only for smaller codeword length. The parameters such as maximum iterations, code length, encoding techniques etc., are analysed during simulations. In both Gallager and progressive edge growth constructions multi- SDBF perform well when comparing to other algorithms.

Normally BER of around 10^{-5} is preferred for analysing SNR since it optimal.

The multi-SDBF achieves better convergence for larger codeword length and for larger error as it depends on the inversion threshold. The single-SDBF can be best used for smaller codeword as it provide good BER performance.

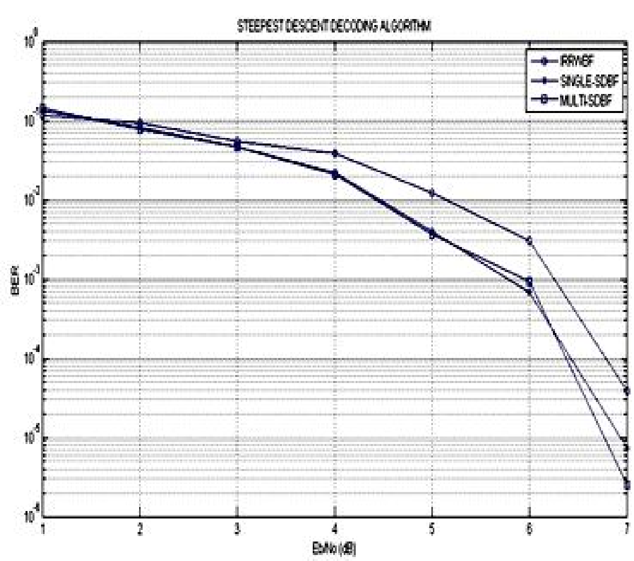

Fig.5: Bit Error Rate comparison of the proposed algorithm Adaptive Multi-SDBF and Single-SDBF against IRRWBF using PEGReg504x1008

The decoding performance of IEEE 802.16e using SDBF algorithm obtained from computer simulations are discussed further. The simulation is done in MATLAB version 7.9.0.529(R2009b). The simulation is analysed for binary 337x672 IEEE 802.16e standard matrix with sub matrix size 28. The resultant matrix is in non-systematic form. The codeword is transmitted through the AWGN channel using bpsk

DOI: 01.IJRTET.9.1.18

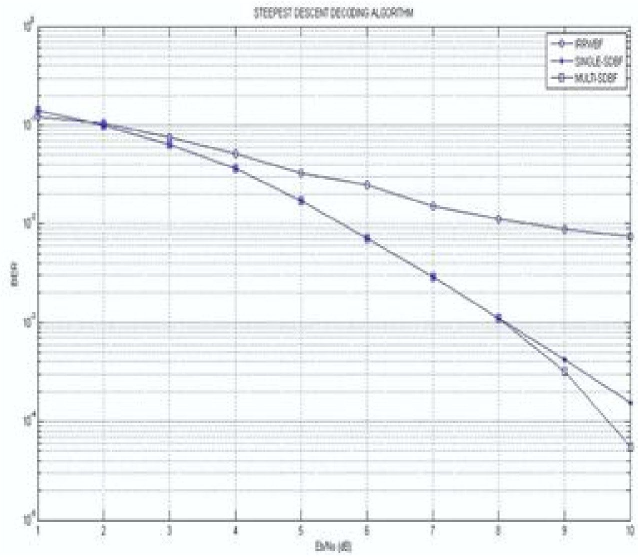

Fig.6: Bit Error Rate comparison of the proposed algorithm using Adaptive Multi-SDBF and Single-SDBF against IRRWBF using Gallager Construction

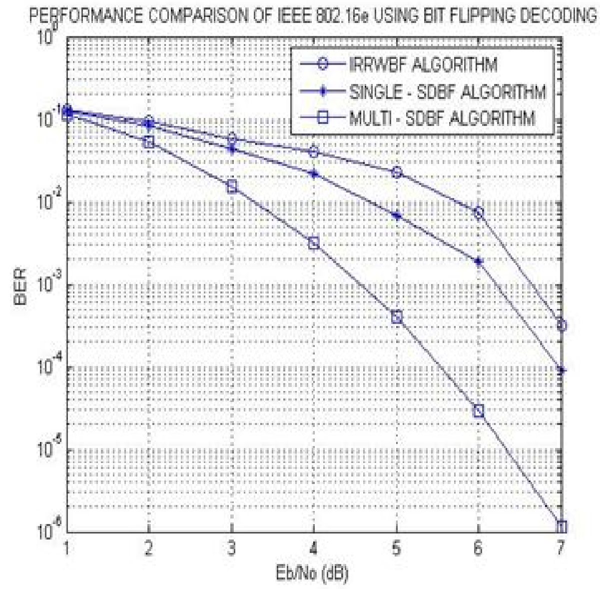

Fig.7: Bit Error Rate comparison of Multi-SDBF, Single-SDBF and IRRWBF for IEEE802.16e

TABLE I :PERFORMANCE ANALYSIS OF DIFFERENT ALGORITHM

H-matrix Size	Encoding Technique	Decoding Technique	Maximum Iterations	Code length	Code Rate	BER	SNR (dB) (Required to achieve the BER)
504x1008	Progressive Edge Growth (Regular)	IRRWBF	20	1008	0.5	10^{-5}	6.9
		Single – SDBF	20	1008	0.5	10^{-5}	6.8
		Multi – SDBF	20	1008	0.5	10^{-5}	6.6
	Gallager Construction	IRRWBF	20	1008	0.5	10^{-2}	9
		Single – SDBF	20	1008	0.5	10^{-2}	5.7
		Multi – SDBF	20	1008	0.5	10^{-2}	5.7

modulation scheme. The maximum decoding iteration is set to 10. The size of the codeword is taken as 672 with a rate 0.5.

Normally better BER performance of around 10^{-5} can be achieved in SPA and MSA algorithm. But the complexity involved in it is very high. SDBF algorithm has very less complexity when comparing to other BF algorithms as well as other soft decision algorithms. Fig 7 shows BER versus SNR for IEEE802.16e for z-factor 28. Simulated results shows a considerable coding gain of around ~0.3 is achieved in the case hard decision algorithms. The SDBF holds good in terms of complexity, thereby providing a tradeoff between complexity and BER performance.

TABLE II: PERFORMANCE ANALYSIS OF DIFFERENT ALGORITHM FOR IEEE 802.16E

Decoding Algorithm	n (bits)	z factor	SNR required to achieve BER of 10^{-5} R=1/2 Max iterations 10	Max iterations 20	Max iterations 30
Single- SDBF	576	24	7.5	8.1	7.7
	672	28	7.5	~8	7.5
	768	32	7.2	~8.5	7.4
	864	36	7.4	~8	7.6
	960	40	7.6	8.1	>8
IRRWBF	576	24	>8	>8.5	>8.5
	672	28	>8	>8.5	~8.3
	768	32	>8	~8.3	~8.3
	864	36	>8	>8.5	>8.5
	960	40	>8	~8.3	~8.3
Multi- SDBF	576	24	6.4	6.3	~6.3
	672	28	6.3	6.3	6.28
	768	32	6.3	~6.1	~6.1
	864	36	6.4	~6.2	~6.1
	960	40	6.3	~6.3	~6

DOI: 01.IJRTET.9.1.18

TABLE III: PERFORMANCE ANALYSIS OF DIFFERENT ALGORITHM FOR IEEE 802.16E WITH Z-FACTOR AS 28

Decoding Algorithm	n (bits)	z factor	SNR Required to achieve BER of 10^{-5} R=1/2 Max iterations 10	Max iterations 20	Max iterations 30
Single-SDBF	672	28	~>7.5	~8	7.5
IRRWBF	672	28	~>8	~>8.5	~8.3
Multi-SDBF	672	28	6.3	6.3	6.28

The Table 2 provides the comparison between different algorithm such as IRRWBF, Single-SDBF, Multi-SDBF for IEEE 802.16e. From the Table 2 it is evident that for a BER of 10^{-5} the SNR required for multi-SDBF is less among hard decision algorithms.

The computational complexity and the processing platform used determines the running time required by each algorithm. Table 3 gives the analysis for different algorithms considering z-factor 28. Table 4 shows the computational complexity of the algorithms in terms of the number of additions and multiplications needed per iteration. In the table, LDPC codes are assumed to be regular. This means that the sizes of sets of non-zero elements in any row and in any column of **H** are constant as $|N(i)|= N$ and $|M(j)|= M$, respectively.

TABLE IV: COMPUTATIONAL COMPLEXITY OF EACH ITERATIONS

ALGORITHMS	NUMBER OF ADDITIONS	NUMBER OF MULTIPLICATIONS
IRRWBF	n(M-1)+m(N-1)	m(N+2)
Single - SDBF	nM+m(N-1)	m(N-1)+n+2nM
Multi - SDBF	nM+mN+2n-3	m(N-1)+ 2n+2nM+ 2

V. CONCLUSION

In this paper, the bit error performance of regular LDPC for single and multi – SDBF mode is analysed. The error and convergence performance is analysed by introducing the additional term to the bipolar syndrome. This scheme uses a similar error-term evaluation method to other algorithms, hence imposing no complexity increase during the iterative decoding process .The inversion threshold is determined by the mean of the negative inversion and variance of the received signal. It is evident from simulation results that Multi-SDBF achieves coding gain of 0.1~0.2 dB over Single-SDBF and IRRWBF for PEG 1008x504.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," in Research Monograph Series. Cambridge, MA: MIT Press,1963.

[2] D. Mackay et.al. "Near Shannon limit Performance of Low density parity check Codes , " Electronics Letters ,Vol.32 no.18 pp. 1645-1646, August 1996.

[3] J. Zhang and M. P. C. Fossorier ," A Modified weighted bit-flipping decoding of low-density Parity - check codes ," IEEE Communications Letters, vol. 8, no. 3, pp. 165–167, Mar. 2004.

[4] M. Jiang, C. Zhao, Z. Shi and Y. Chen, "An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes," IEEE Communications Letters, vol. 9, no. 9, pp. 814– 816, Sep. 2005.

[5] F. Guo and L. Hanzo, "Reliability ratio based weighted bit-flipping decoding for low-density parity-check codes," Electronics Letters, vol.40,no. 21, pp. 1356–1358, Oct. 2004.

[6] T.Wadayama , K. Nakamura , M. Yagita ,Y. Funahashi , S. Usami , and I.Takumi , " Gradient descent bit flipping algorithms for decoding LDPC codes," IEEE Transactions on Communications , vol. 58,no. 6, pp.1610 – 1614, Jun.2010.

[7] D. J. C. Mackay, " Encyclopedia of sparse Graph codes". [Online] , Available : http://www.inference.phy.cam.ac.uk/mackay/odes/data.html

[8] IEEE 802.16eÔ-2005: Part 16: "Air Interface for Fixed and Mobil Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1"

[9] Daud, M. ; Suksmono, A.B Hendrawan, Sugihartono ,"Comparison of decoding algorithms for LDPC codes of IEEE 802.16e standard " Telecommunication Systems, Services, and Applications (TSSA), 2011 6th International Conference on Digital Object Identifier pp. 280-283.

[10] P. Jagatheeswari , M. Rajaram, "Performance Comparison of LDPC Codes and Turbo Codes ", European Journal of Scientific Research , Vol.54 No.3 (2011), pp.465-472.

DOI: 01.IJRTET.9.1.18